Introduction to Computing — Lab

Week 06

Topic

Loops (while, for, do while)

Objective

- To improve the readability by clearly defining different outcomes based on specific conditions.
- To understand complex decision-making by evaluating multiple conditions.
- To reduce code duplication and improve maintainability by using loops for repetitive tasks.
- To provide mechanisms to control the number of iterations **(e.g., for, while, do-while).**

Outcomes

1. Facilitate complex decision-making processes, allowing for detailed evaluations of conditions within iterative structures.
2. Provide a clear and organized way to manage multiple conditions and repetitive tasks, improving code clarity and flow.
3. Enable programs to respond dynamically to varying inputs and states, adjusting behavior based on both conditions and iterations.

Content

The following topics require understanding for this lab session:

1. Loops (for)

Details of these topics are given below:

1. Loop

Loop in programming is a control structure that allows for the repeated execution of a block of code as long as a specified condition is met. Loops are essential for automating repetitive tasks and can iterate a fixed number of times or continue until a certain condition becomes false. In C++, common types of loops include:

- **while**
- **do-while**
- **for**

Tasks

Students are required to complete the following tasks in lab timings.

Task 1

Write a C++ program that repeatedly prompts the user to enter a number until they enter a number greater than 10. Once a valid number is entered, display that number.

Example Output:

```
Enter the Number: 5
Invalid Number/ Less than 10.
Please Enter Number Again

Enter the Number: 20
Yes, Valid Input, Number is 20
```

Task 2

Write a program that inputs a number and checks whether it is a perfect number or not. A perfect number is a number that is numerically equal to the sum of its divisors.

For example, 6 is a perfect number because the divisors of 6 are 1,2,3 and 1+2+3 = 6

Example Output:

```
Enter a number: 6
6 is a perfect number
```

Task 3

Write a C++ program that prompts the user to enter positive numbers and calculates their sum. The program should continue to accept numbers until the user enters a negative number, at which point it should display the total sum.

Example Output:

```
Enter the Number: 5
Enter the Number: 4
Enter the Number: 7
Enter the Number: 2
Enter the Number: -9

        Sum of positive Number is: 18
```

Task 4

Write a C++ program that counts down from a specified number to zero. The program should prompt the user to enter the starting number and then display each number in the countdown. The countdown should continue until it reaches zero.

Example Output:

```
Enter the Starting Number: 10
Count down is: 10 9 8 7 6 5 4 3 2 1 0
```

Task 5

Write a C++ program that prompts the user to enter a series of integers. The program should count how many of the entered numbers are even and how many are odd. The program should continue to accept numbers until the user enters zero, at which point it should display the counts.

Example Output:

```
Enter the Number :1
Enter the Number :2
Enter the Number :5
Enter the Number :7
Enter the Number :8
Enter the Number :9
Enter the Number :10
Enter the Number :0

        Total even Numbers: 3
        Total Odd Numbers: 4
```

Task 6

Write a program using for loop in which the user enters a number **X** and your program should calculate the sum of all the numbers till Number entered by user

Expected Output

```
Enter a number: 5
Sum is: 15
```

Task 7

Create a simple calculator program that allows the user to perform addition, subtraction, multiplication, or division. The program should continue to prompt the user for operations until they choose to exit.

Expected Output

```
Enter 1st Number: 5
Enter 2nd Number: 2

    --Press 1 for Addition –
    --Press 2 for Subtraction --
    --Press 3 for Multiplication --
    --Press 4 for Division –
    --Press 0 for exit --

Enter Your Choice: 3
Result is 15
```

Task 8

Write a C++ program that calculates the factorial of a positive integer (N) entered by the user..

Example Output:

```
Enter the Number: 4
Factorial is :24
```

Task 9

Write a C++ program that prompts the user to enter a series of numbers and identifies the largest number entered. The program should stop accepting numbers when the user enters a negative number.

Expected Output

```
Enter Number: 89
Next Number: 78
Next Number: 70
Next Number: 17
Next Number: 85
Next Number: -92
Largest Number is: 89
```

Task 10

Write a program to calculate and display the following series and then the sum of the following polynomial series using do while loop. $\mathbf{1 + 2x + 3x^2 + 4x^3 + 5x^4 + 6x^5}$

Expected Output:

```
Enter a value of x: 2
Enter degree: 5
1 + 4 + 12 + 32 + 80 + 192 = 321
```